

Reti Logiche T

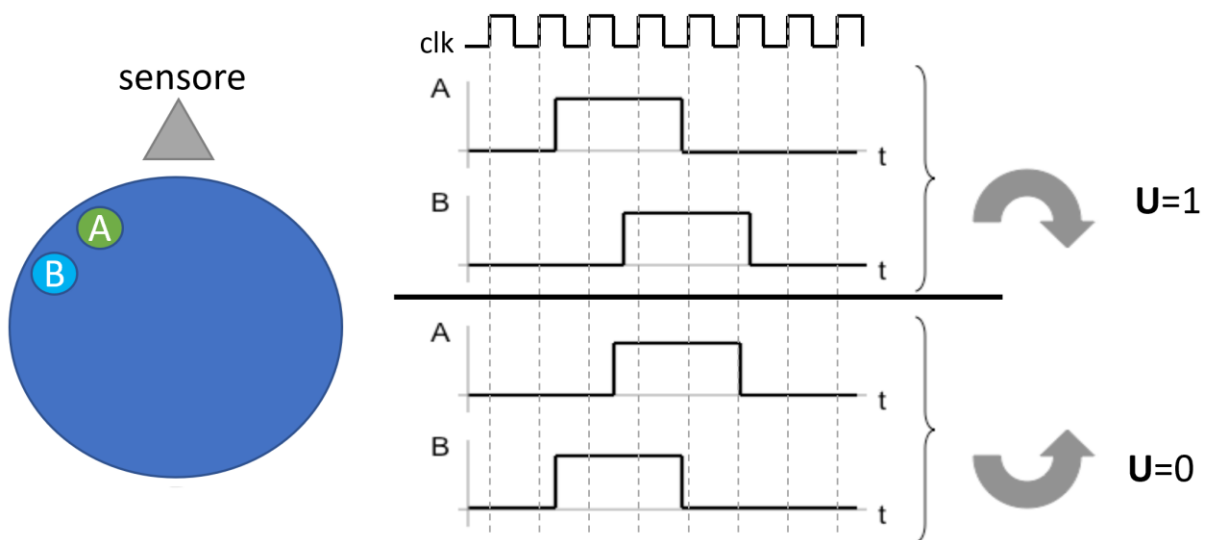
Esercizi reti sequenziali sincrone a sintesi diretta

Esercizio 1

Progettare in **modo diretto** una rete sequenziale **sincrona** in grado di individuare il **senso di rotazione** (**orario** o **antiorario**) di un disco che ruota.

Il sistema presenta un sensore capace di rilevare due tag, denominati **A** e **B**, piazzati opportunamente sul disco: quando uno di questi tag passa sotto al sensore, questo mette a 1 l'ingresso della rete chiamato come il tag rilevato (**A** o **B**).

Il sensore è progettato in modo da generare le due forme d'onda mostrate nella figura, rispettivamente durante la rotazione oraria e antioraria (notare la sovrapposizione tra gli intervalli in cui i due segnali assumono il valore 1). Si assuma che la durata di un periodo del clock utilizzato sia sensibilmente inferiore al lasso di tempo per cui i segnali assumono il valore 1.



L'uscita **U** della rete indica se il disco ruota in **senso orario** (**U=1**) o in **senso antiorario** (**U=0**). L'uscita **U** deve codificare l'ultima misura effettuata, fintantoché non è possibile eseguire un aggiornamento. Il segnale **A_RESET** consente di inizializzare la rete in modo asincrono nel momento opportuno.

Si proponga una soluzione basata su un **approccio sincrono**, riducendo al minimo l'utilizzo delle risorse.

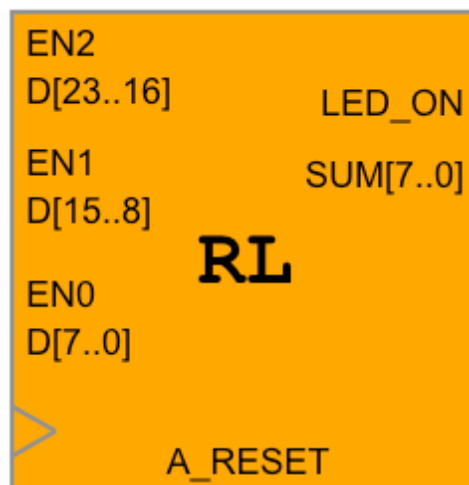
Esercizio 2

Progettare **in modo diretto** una rete sequenziale sincrona con 3 ingressi **D[7..0]**, **D[15..8]** e **D[23..16]** e due uscite denominate **LED_ON** e **SUM[7..0]**.

A ciascun ingresso **D[7..0]**, **D[15..8]** e **D[23..16]** è associato un segnale di *enable* (rispettivamente **EN0**, **EN1** ed **EN2**) che qualifica l'ingresso come valido (1) o non valido (0). Dall'avvio, successivo a un'inizializzazione mediante un ulteriore ingresso **A_RESET** che agisce in modo asincrono, la rete deve **continuamente accumulare**, modulo 256, prima **tre caratteri validi da D[7..0]**, poi **1 carattere valido da D[15..8]** e infine **2 caratteri validi da D[23..16]** e aggiornare, nel clock successivo alla ricezione di ogni carattere valido, il risultato dell'operazione di accumulo sui segnali **SUM[7..0]**.

Il ciclo di letture dai 3 ingressi si ripete continuamente con l'ordine descritto in precedenza tranne quando **SUM[7..0]** risulta *palindromo*. Ogni volta che tale circostanza è verificata, nello stesso ciclo di clock in cui **SUM[7..0]** diventa palindromo deve essere **invertito il valore di LED_ON** (all'avvio impostato a 1) e la lettura successiva deve essere eseguita da **D[15..8]** per poi proseguire in accordo a quanto indicato in precedenza (ovvero, dopo la lettura da D[15..8] dovranno seguire due letture da D[23..16], tre letture da D[7..0], una lettura da D[15..8], due letture da D[23..16], eccetera) fino alla ricezione del successivo carattere valido che rende palindroma l'uscita.

Tutti i segnali sono sincroni. Il segnale **A_RESET** agisce in modo asincrono inizializzando, all'avvio, la rete. Eseguire la sintesi della rete evidenziando e risolvendo eventuali criticità e minimizzando l'uso delle risorse.



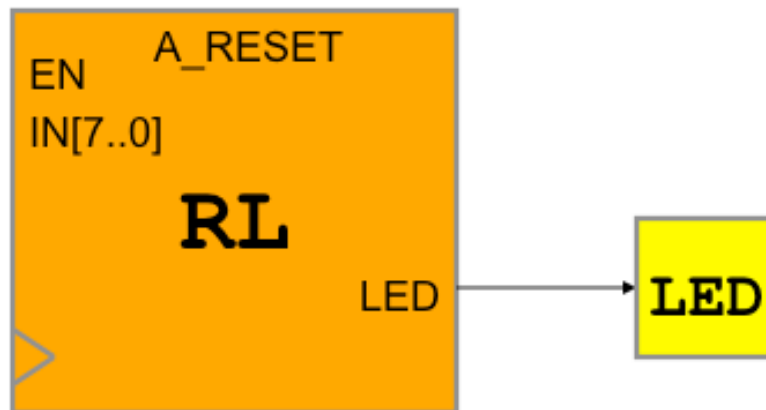
Esercizio 3

Progettare **in modo diretto** una rete sequenziale sincrona che controlla continuamente i caratteri **IN[7..0]** in ingresso quando **EN=1**. Allorché **il carattere in ingresso sia divisibile per 16 e il precedente carattere valido sia divisibile per 32** la rete deve invertire, al termine della ricezione del carattere divisibile per 16, **lo stato di un led inizialmente acceso**.

E' presente inoltre un segnale, denominato **A_RESET**, che consente di inizializzare la rete in maniera asincrona; tutti gli altri segnali sono da intendersi sincroni.

Si consideri il valore 0 non divisibile per 16 o 32.

Eseguire la sintesi della rete evidenziando e risolvendo eventuali criticità e minimizzando l'uso delle risorse.



Esercizio 4

Progettare **in modo diretto** una rete che monitora continuamente i 3 segnali di ingresso **A, B, C**. I segnali sono da intendersi validi solo quando **EN=1**. La sequenza da monitorare è la seguente: **A=1, B=0, C=0, B=1, C=1** e così via (ovvero, la sequenza successiva da rilevare sarà nuovamente **A=1, B=0, C=0, B=1, C=1**, ecc). Anche se in ogni periodo di clock sono disponibili tutti e 3 i segnali, la rete attende come prima cosa un periodo di clock con **EN=1** in cui **A=1** (ignorando il valore di B e C in quel clock), poi attende il clock successivo in cui **EN=1** e **B=0**, e così via.

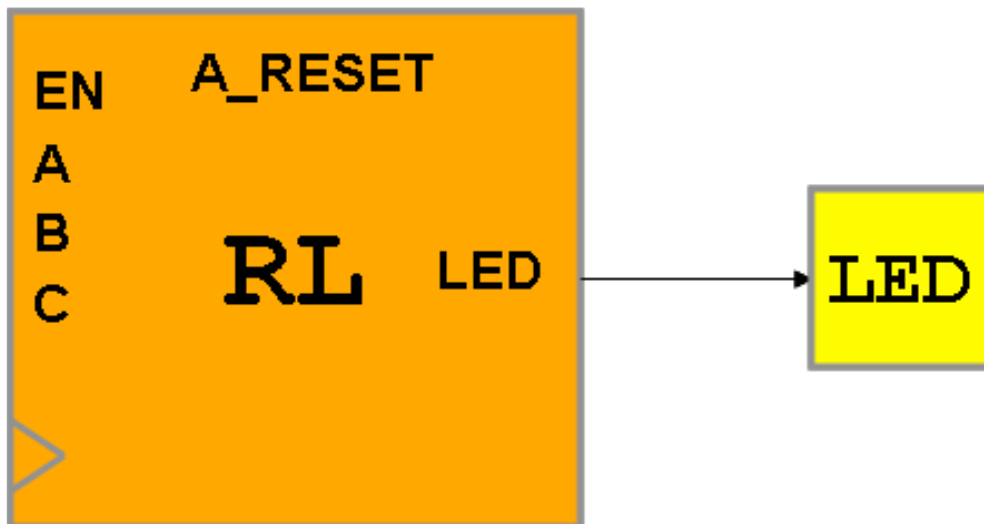
Nel caso sia ricevuto, con **EN=1**, un valore diverso da quello atteso la rete deve rimanere in attesa del valore corretto **senza ripartire** con l'analisi del primo valore della sequenza. Al termine della ricezione di ogni sequenza completa deve essere invertito lo stato di un **LED** inizialmente spento.

Tutti i segnali di ingresso sono sincroni. Il sistema è inizializzato all'avvio mediante il segnale asincrono **A_RESET**.

Eseguire la sintesi della rete evidenziando e risolvendo eventuali criticità e minimizzando l'uso delle risorse.

Determinare l'espressione della massima frequenza di clock della rete considerando esclusivamente la generazione del segnale **LED**, assumendo che:

- Il ritardo di propagazione nei Flip Flop sia di 10 ns
- Il tempo affinché i segnali d'ingresso siano stabili sia di 12 ns
- Il ritardo di propagazione nei gate sia di 1 ns
- Il tempo di setup dei Flip Flop sia di 2 ns.



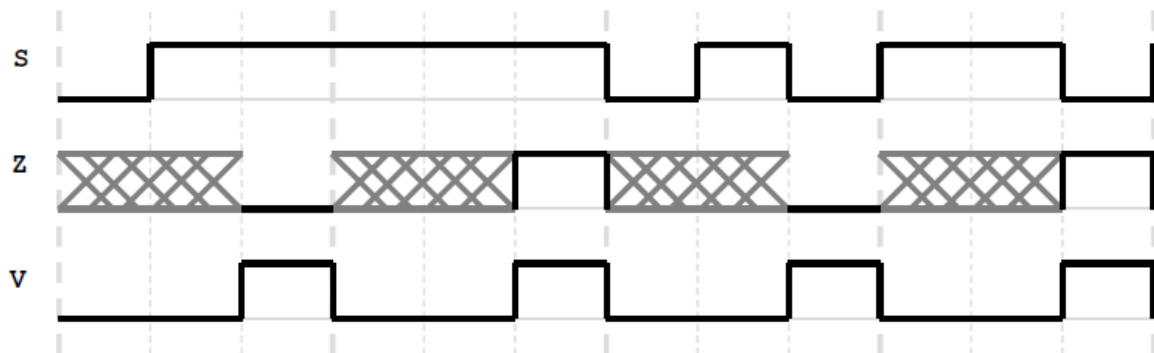
Esercizio 5

Una RSS deve monitorare i bit ricevuti da una linea seriale **S** al fine di individuare, all'interno di **gruppi di 3 bit consecutivi**, valori binari minori di 2 e maggiori di 5.

Ogni bit ricevuto dalla linea seriale deve essere associato ad un solo **gruppo di 3 bit** (ovvero, i gruppi sono contigui temporalmente ma non hanno elementi in comune) e i bit all'interno di ogni gruppo sono **ricevuti a partire dal bit più significativo**. La RSS, **durante la ricezione del terzo bit** di ogni gruppo, deve indicare sull'uscita **Z** se gli ultimi tre bit ricevuti codificano un numero $N > 5$ o $N < 2$ (in tal caso, $Z=1$) oppure un numero $2 \leq N \leq 5$ (in tal caso, $Z=0$).

La rete deve anche segnalare attraverso l'uscita **V** che è **in corso la ricezione del terzo bit del gruppo** ($V=1$); l'uscita **Z** è significativa solo quando $V=1$.

Eeguire la sintesi della rete evidenziando e risolvendo eventuali criticità e minimizzando l'uso delle risorse.



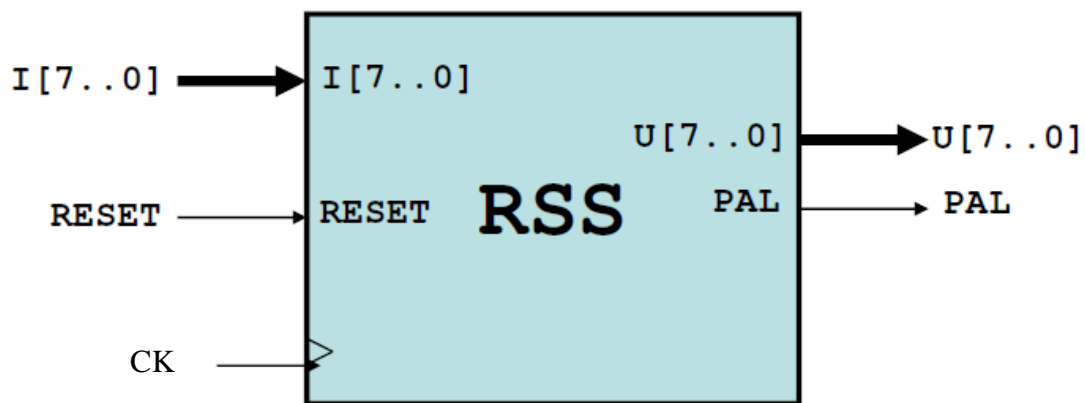
Esercizio 6

Una RSS deve continuamente monitorare i caratteri ricevuti attraverso le 8 linee **I[7..0]**. Allorché viene ricevuto il carattere **0x46** la RSS, dal successivo carattere, inizia a contare **modulo 256** quanti byte, interpretati come numeri senza segno, sono **maggiori di (63)₁₀**. Il conteggio, fornito in uscita attraverso i segnali **U[7..0]**, dovrà essere interrotto in seguito all'arrivo del carattere **0x3**. Il conteggio riprenderà con le stesse modalità quando verrà ricevuto nuovamente il carattere **0x46** e così via...

Quando il conteggio è abilitato (secondo le modalità appena descritte), e solo in questa circostanza, la rete deve anche segnalare all'esterno mediante il segnale **PAL** quando la stringa che codifica i caratteri ricevuti è *palindroma* (ovvero, assume lo stesso valore se letto da sinistra a destra o da destra a sinistra). Se non si è in fase di conteggio, il segnale **PAL** deve essere mantenuto a "1".

Si preveda infine la possibilità di inizializzare la rete mediante un segnale di **RESET sincrono**, prioritario rispetto a tutti gli altri segnali anch'essi sincroni.

Eseguire la sintesi della rete evidenziando e risolvendo eventuali criticità e minimizzando l'uso delle risorse.



Esercizio 7

Progettare una RSS in grado di generare (in base a due bit di programmazione **X1** e **X0**) le seguenti quattro sequenze di uscita a 2 bit (segnali **U1** e **U0**):

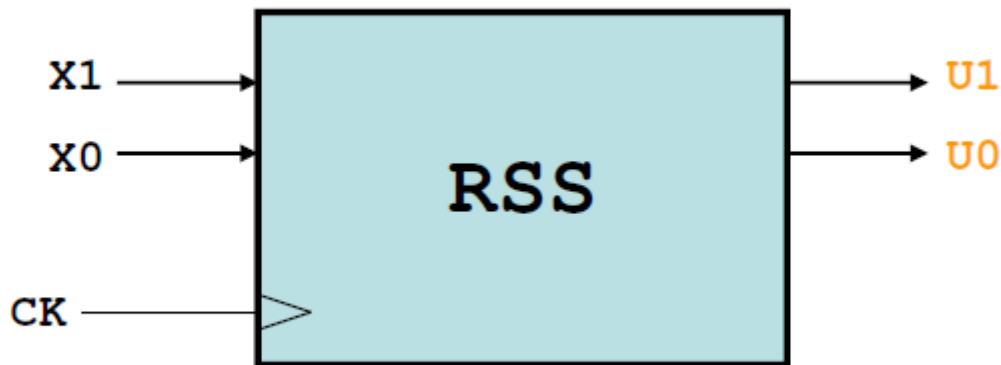
- **10 01 01 10 11** se **X1=0** e **X0=0**
- **10 01 10 01 10 10 11** se **X1=0** e **X0=1**
- **00 10 00 11** se **X1=1** e **X0=0**
- **01 10 10 10 10 11** se **X1=1** e **X0=1**

I segnali di ingresso **X1** e **X0** sono significativi solo quando l'uscita della RSS vale **11**. Pertanto, la determinazione della sequenza da generare deve essere stabilita solo quando l'uscita della RSS vale 11.

In tutti gli altri momenti i segnali X1 e X0 assumono valori indefiniti e privi di significato ai fini della corretta generazione della sequenza di uscita.

Il sistema è inizializzato all'avvio mediante il segnale asincrono **A_RESET**.

Eseguire la sintesi della rete evidenziando e risolvendo eventuali criticità e minimizzando l'uso delle risorse.



Esercizio 8

Progettare **in modo diretto** una rete sequenziale **sincrona** in grado di analizzare un flusso dati continuo a 8 bit sui segnali di ingresso **IN[7..0]**. Tali segnali, sono da considerarsi significativi solo quando il segnale **EN** vale 1.

Specificatamente, la rete da progettare dovrà fornire **su uscite indipendenti** le seguenti informazioni:

- a) numero di byte di ingresso validi e divisibili per 64 (zero escluso)
- b) numero di byte validi e palindromi
- c) numero di volte che a un byte valido pari segue un byte valido dispari.

Tutti i conteggi dovranno essere eseguiti **modulo 256**. A tal proposito è possibile utilizzare come elemento di base **un contatore modulo 16** dotato di **ENABLE sincrono** e **RESET asincrono**. Si consideri il contatore modulo 16 già progettato.

Tutti i segnali d'ingresso sono sincronizzati con il clock. Inoltre, nel sistema è presente un ulteriore segnale **A_RESET** che consente di inizializzare la rete in modo asincrono.

- Evidenziare e risolverle eventuali criticità
- Ridurre al minimo l'utilizzo di risorse.

Esercizio 9

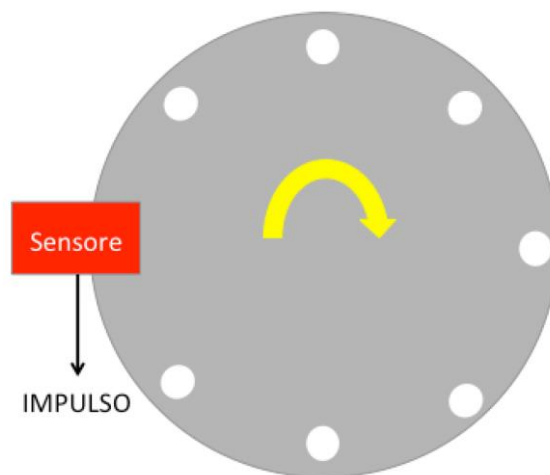
Progettare **in modo diretto** una rete sequenziale **sincrona**, con un segnale clock a 4 KHz, in grado di misurare gli impulsi rilevati in 1 sec da un opportuno sensore, durante la rotazione di un disco circolare con 8 fori, disposti in modo equidistante lungo la circonferenza del medesimo disco come mostrato nella figura.

Il sensore, fermo rispetto al disco che ruota senza mai invertire il proprio moto, genera un segnale **IMPULSO=1** quando il foro si trova nell'area monitorata e **IMPULSO=0** in caso contrario (ossia quando nell'area controllata dal sensore non si trova il foro presente sul disco).

La rete da progettare deve aggiornare, ogni volta che passa un secondo, il numero di impulsi rilevati durante il secondo appena trascorso. Tale numero deve essere codificato in binario come intero senza segno e mostrato stabilmente sul bus di uscita **N[K..0]**. Si supponga che il disco esegua non più di 5 rotazioni ogni secondo.

Infine, la rete da realizzare deve presentare un ingresso **A_RESET** per resettare in modo asincrono il valore mostrato su **N[K..0]**.

- Indicare quanti bit **K** sono necessari per codificare la misura
- Evidenziare eventuali criticità e risolverle in modo appropriato
- Ridurre al minimo l'utilizzo di risorse.



Esercizio 10

Progettare **in modo diretto** una rete sequenziale **sincrona** con due ingressi **EN** e **D[7..0]** e un'uscita denominata **DETECTED**.

La rete deve analizzare di continuo i numeri (codificati come interi senza segno) ricevuti su **D[7..0]**, considerando validi solo i numeri ricevuti quando **EN=1**. La rete deve portare l'uscita **DETECTED** ad **1**, esattamente per **tre** clock, al termine della ricezione della seguente sequenza di caratteri: **4, 6, 9, 11, 2** (più precisamente dal clock successivo a quello in cui è stato ricevuto il numero **2**).

L'eventuale ricezione di caratteri con **EN=1** non appartenenti all'ordine indicato deve essere ignorato (ad esempio nella sequenza **4-6-5-1-9-8-11-2**, i caratteri **5, 1 e 8** vanno ignorati, proseguendo con l'analisi della sequenza ogni volta che si riceve uno dei caratteri d'interesse). Tuttavia, la ricezione del carattere **68**, con **EN=1**, deve fare ripartire l'analisi della sequenza dal primo carattere (ossia **4**).

Inoltre, al termine della ricezione di ogni sequenza d'ingresso valida, la rete deve immediatamente riprendere l'analisi della sequenza successiva mentre genera l'uscita **DETECTED** in accordo a quanto indicato in precedenza. Tutti i segnali d'ingresso sono sincroni e all'avvio è asserito un ulteriore segnale, denominato **A_RESET**, che consente di inizializzare la rete in modo asincrono.

- Esistono elementi di criticità? In caso affermativo descriverli chiaramente e risolverli.
- Ridurre al minimo l'utilizzo di risorse.